International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 6 Issue: 5 17 - 26

Java Zoo a Tool for Interactive Java Instruction

Cheryl D. Seals, Akolade Asipade, Yang Cao, Robertson Bassy

Abstract: Games have been a form of entertainment for centuries and used to focus and motivated which is a great feature that can be utilize games as the basis for creating game-based educational applications. Games are part fun and part leisure, trying to educate using games is very interesting, but care must be taken in designing the game interfaces as they should be simple, attractive, usable, and at the same time portray the concept of learning. Many students have found Computer Science as a very challenging subject and some of the most challenging for some new CS1 students is their first programming language. We lose many students to other majors at the level of CS1 and can weed out potential great future computer scientists.
In our computing department, Java is the first programming language and many students have problems with it because they do not understand the concepts of classes, objects, encapsulation, inheritance and polymorphism. It has been found that games motivate students and focuses them for long period of time. With long periods of time being required to have basic functional understanding of programming language, games can play an important role for intrinsic motivation. We propose a Java game called "JavaZOO" to demonstrate the concepts of programming by mapping the animal kingdom to the object-oriented design principles.

*****

1. Introduction

One of the problems we are aiming to address is student retention. During the initial programming classes, computer science loses many students to other majors as learning their first programming language can be a very challenging task for many students. In many cases, CS1 (java) hasbeen a bottleneck to entering the major.Java is the first programming language and many students have problems with it because they do not understand the relation between the concepts of classes, objects, encapsulation, inheritance and polymorphism in Java. It has been found that games motivate students and focuses them for a long period.

The proposed Java game called "JavaZOO" demonstrates the concepts of programming by mapping the animal kingdom to the object-oriented design principles. The actions and the flow of the game reinforce the concepts of the programming language. JavaZOO will cover the concepts of classes, objects, encapsulation, loops, inheritance and polymorphism

1.1 Why do we learn to program?

The computer is one of the most powerful innovations in human history. With the use of computers, people are suddenly able to perform a staggering amount of computations at dazzling speeds. Information can be crunched, organized, and displayed in the blink of an eye. As technology continues to advance, the computer will become even more pervasive and small. Millions of devices that we see around us everyday use some or other form of computation.

With the computer being a major force that drives our daily lives, We need more individuals to become masters of the technology and learn to program. When you program, you are a creator.

Programming lets you work with your mind to free yourself to create things that are interesting or useful instead of being limited by the confines of what others have done before, and even when using those other things, you'll find that you can better appreciate and understand them [6].

1.2 How to learn to program

Over the years, researchers have devised different theories and ways that help people learn to program. In a study, students were interviewed about their understanding of what learning to program means. Many students talked about learning to program in terms of learning a special way to think, different from other subjects studied. Many of these students had problems in describing what this special way to think included.

Since people are not born as programmers, it becomes extremely important to follow and pay a lot of attention to how this knowledge of interacting with computers is given to a person.

Bransford, JD in his book "Human Cognition" explains, "Meaningful learning", as a process in which the learner connects new material with the knowledge that already exists in the memory. The human cognitive system can be divided into short-term memory and long-term memory. When new technical information enters the human cognitive system from the outside, it should go through series of steps for meaningful learning to occur. The steps include

1. Reception – Learner should pay attention to the incoming information so that it reaches short-term memory.
2. Availability – Learner must possess appropriate prerequisite concepts in long-term memory to use in assimilating the new information.

3. Activation – Learner must actively use this prerequisite knowledge during the learning so that the new material may connect with it.

According to Bransford, if any of these steps are not followed or conditions are not met, meaningful learning cannot occur; and the learner will be forced to memorize each piece of new information as a separate item to be added to the memory.

In our research, we want to connect our application to a set of knowledge that already exists in their long memory. As learners, we create semantic networks of association and if we connect our next application to existing long term memory through semantic connections this may increase the effectiveness of our educational game. The proposed educational game will explore this concept and attempts to provide new knowledge of programming language using existing knowledge of animals, zoo and the division of animal kingdom in the brain of the users. The concepts of Object – Oriented programming are explained on the basis of division of classes in the animal kingdom classification. The attempt is to be able to gain the learners interest, so that the reception of information occurs and then availability and activation of information can help in learning of the programming language concepts.

1.3 Models for teaching and learning programming

Researchers have studied student-generated programs, and found that learners had more trouble putting the pieces of a program together than learning the programming language concepts [1]. Some of these models include:

- Semiotic ladder
- Cognitive objectives taxonomy
- Problem-solving

1.4 Learning with JavaZOO

As discussed in the previous sections, concepts of programming languages can be reinforced with the help of teaching models. The proposed Java game JavaZOO will aid CS1 students in learning Java, which has been a bottleneck to entering the major. JavaZOO will use animal kingdom hierarchy to relate to the object-oriented concepts. The Java game will increase the interest and learning of programming concepts as students will be able to relate to real-life examples. They will enjoy the graphics and animations of the game, which can be a feature for keeping them interested in the game. Next few chapters discuss in detail about the proposed Java game and its features.

2. Literature Review

Over the years, it has been an important area for researchers to devise and study new methods to effectively aid in learning of programming or computation by novice learners. The techniques fall into a broad range including traditional lecture-based instruction, hands-on coding, problem-solving, participating and enculturation, use of media and animation, use of text and graphics to enhance learning materials, etc. Next, we will discuss a few techniques proposed by different researchers.

2.1 Learning Process

In her study of introductory students, Booth [12] found that learners conceived learning programming in four ways, with the following structure:

• Learning a programming language
• Learning to write programs in a programming language
• Learning to solve problems in the form of programs
• Becoming part of the programming community

While the two former stages concern coding skills, the third one requires that the student is also able to apply the skill in the appropriate manner when confronted with a problem not being expressed in programming terms. “Becoming part of the programming community,” means, (e.g. considering self as one member of a programming team that also interacts with users).

When starting to learn programming, learners make a mistake of using a programming language as if the computer makes logical assumptions. Since the students had little understanding of how computer operate, they apply the human principles of thinking and acting to the computer. We may therefore say that the first step of learning programming is going from assumptions that the computer will make logical inferences to understanding that we must learn programming as learning any other language. This includes understanding programming concepts and the way programs are executed, in terms of internal variables, external files, and I/O.

2.2 Using Games

In our research, we found that learning is more enjoyable when we infuse learning experienceswith educational. This certainly does not mean that the text and other media representations should not be used, but using games with the other media can provide a self-motivated and interested learner. We found that in coming years more and more people will be interested in games, more and more people will play games and the educational game industry will be a great success [8]. We believe that if games continue to follow the strong learning principles that keep the learners interested in learning the difficult and complex modes in the game even when they are challenging and time-consuming, more and more people will learn through games in coming years. Z. Fang [8] in his paper lists several benefits of games and graphics over text-only representations: He summarizes that game can support learning in the following ways:

- Develop the characters

- Develop the plot
- Provide a different viewpoint
- Motivate the learner
- Promote creativity
- Serve as mental scaffold
- Foster aesthetic appreciation
- Promote language and literacy
- Aid remembering

There are several empirical studies that have examined the effects of game-based instructional programs on learning. For example, both Whitehall and McDonald [13] and Ricci [11] found that instruction incorporating game features led to improved learning. Games promoted increased risk-taking among students, which resulted in greater persistence on the task and improved performance. Instruction that incorporated game features enhanced student motivation, which led to greater attention to training content and greater retention. Pierfy [10] evaluated the results of 22 simulation-based training game effectiveness studies to determine patterns in training effectiveness across games. Twenty-one of the studies collected learning data that generally consisted of paper-and-pencil fact-and-principle knowledge tests. Three of the studies reported results favoring the effectiveness of games over conventional teaching; three reported results favoring the effectiveness of conventional teaching over games; and the remaining 15 found no significant differences. Eleven studies also tested retention of learning. Eight of these studies indicated that retention was superior for game-based training; the remaining three yielded no significant differences. Level of student preference for training games over classroom instruction was assessed in 8 of the studies, and in 7 of those studies, students reported greater interest in simulation game activities than in conventional teaching methods [10]. More recently, it was concluded that games seem to be effective in enhancing motivation and increasing student interest in subject matter, yet the extent to which this translates into more effective learning is less clear [9].

3. Methodology

3.1 Motivation

Games have been a form of entertainment for centuries. Games keep the user motivated and focused throughout the life of the game. These features are the rationale to utilize games as the basis for creating game-based educational applications. Games are part fun and part leisure, trying to educate using games is very interesting, but care must be taken in designing educational game interfaces as they should be simple, attractive, usable, effectively support learning. There is not much support available for the educational use of games in our society as most games are thought of as merely medium of entertainment.Although most researchers agree that games can be engaging and instructive, there is little consensus regarding the use of games for educational purposes [24]. Research has shown that if we integrate educational content with the game features, we can inculcate serious motivation in the users to learn the proposed educational materials. Advancement in the technology has provided easy access to the Internet on mobile phones. The Internet on mobile devices has broken traditional boundaries and made it possible for learning to happen anywhere, anytime and in any environment, but there are several issues that arise when combining games with education, especially in the context of mobile interfaces based on the design principles of PC games which in some cases cannot be migrated to mobile games [24].

3.1.1 Description of initial study

The initial study for the design and development of this educational game included how effectively the concepts of programming can be mapped to our content example of the animal kingdom. Throughout the design and development of the project, several questions were the basis for design requirements.

- Can programming concepts be taught using an educational game?
- How educational games can prove effective learning and teaching?
- How balance can be maintained in an educational game, keeping both factors alive, the game and education.
- What is a good educational game? Studying some existing educational games that teach programming languages.
- How effectively can the mapping of programming concepts and animal kingdom be designed in the form of a game.
- Which object-oriented language should be chosen to be explained during the game?
- A Game should be developed for which platform?
- What are some of the object-oriented concepts that are most difficult for a person to understand and map to the real world?
- Can being able to map object-oriented concepts to the real-world help in programming with better style?
- What should be the game flow?
- How will the concepts be represented and explained, keeping in mind the screen size restriction of a mobile device?
- Should a code sample and quiz be provided or not?
- Graphics play an important role in the game, how should the graphics be designed? What colors should be used to keep the users interested in the games?

- What heuristics should be followed for developing a game for a mobile device?
- Which object-oriented concepts should be included?
- Is the proposed game serving the purpose of teaching the programming concepts to the novice users?
- What should be included in the intermediate testing with low-fidelity prototype?
- How will the game be tested with the users and how the responses should be collected?

All these questions were answered in refining the design and preliminary development. Throughout this chapter, the answers to these questions are provided in detail. These questions played an important role in assuring that none of the sides are left unvisited, that the view of users is considered, that sufficient literature has been reviewed and that a complete development process will be followed.

3.2 Factors about games that rationale their use in education

Jane McGonigal [20] in her book "Reality is Broken" has said that when we are in game worlds, we become the best version of ourselves, most likely to help at a moment's notice, most likely to stick with a problem as long as it takes, to get up after failure and try again. People play games with the goal to win the game; we can witness an intense concentration and deep focus during the game play. They are motivated throughout the game and are willing to retry a number of times. They are full of optimism and believe that they can solve the problem that the game offers. They carry urgent problem solver and agile attitude while playing games. The rewards that games offer in form of levels up, prizes, unlocking new features etc. makes the gamer believe that he can achieve more. The immediate feedback and results of the efforts that are put in by the gamer also plays an important role in keeping the gamer motivated and interested in the game. A game can help a gamer experience a real or a non-real world/platform that he might never be able to experience in real life. This phenomenon makes a person more thoughtful and adventurous. Tzvi Freeman [16] suggests that:

- A good game empowers your imagination
- A good game makes you feel in charge.
- A good game is transparent. You only feel your own mind, the other player and your ideas.
- A good game lets you into its creator's imagination.
- A good game lets its players feel each other's personality.
- A good game fits the human being like a glove.

Research has shown that games and graphics generate a greater stimulus in the brain by which we remember a larger portion of the context than the text representation of the same context [16]. Introduction of graphics prepares a perceivable picture of the context in the brain and helps in understanding and remembering the context.After analyzing all the points the question that arises is "Can games be used for educational purposes"? We believe that the answer is yes, because if the enjoyment and pleasure that games provide can be migrated to the educational world, then, we can create motivated learners. We can create learners who do not give up easily and believe in their ability to solve the proposed problem. All the advantages of the games can be mapped into lifelong learning. However, there is little consensus in use of games for education because learning and education are accepted as serious endeavors in our society and it is believed that serious pedagogy should be followed in a disciplined way to achieve successful education [24].The next section throws light on some statistics related to educational gaming, and explains that many prefer games to the conventional teaching methodologies. The section also describes the issues with educational game development and investigates the use of mobile Interfaces for educational game development.

Heuristics for Mobile Game Development

Several heuristics have been suggested that can help designers in coming up with usable educational games that are developed keeping in mind the issues related to game development and mobile interfaces[24].

- **Adaptation**
  One important characteristic is that games should be able to adapt to the learning curve of the learner. The difficulty level or the information level of the game should increase or decrease gradually per every learner's progress in the game. If games are adaptive they support learner preferences for different access pathways and allow the learner to find relevant information while at the same time remaining immersed in the game [22]. For example, "*Poccer"*, a pocket football game increases or decreases the speed of computer opponent depending upon the player's score and performance.

- **Challenge and mastery**
  Learners with different abilities and varying skills play the games. A good game is one that is easy to learn for a user but the challenges never go away. Researchers identify three types of difficulty curves for games: the flat curve, the linear progression, and the s-curve [24]. In the flat curve, the game complexity does not adapt to the player and traditional pre-determined levels come up during the game play. The levels in the game are mostly of the same difficulty level. In the linear curve, the difficulty level of the game steadily increases with time. This may also be not a very good practice

when visualizing players with different skills. Slow learners might not have enough time to adapt to the different levels of the game and the difficulty would increase linearly. A player gradually might lose interest in the game. In the s-curve, the difficulty level increases slowly and lets the player learn how to play the game. The difficulty level gets steeper during the bulk of the game and in the last two to five hours flattens out, allowing players who make it through most the game to survive through to the finish.

- **Goals**
  The final goal and the sub-goals of the game should be apparent to the user at every point of time in the game. The goals should also be consistent. The generated goals should be meaningful, obvious, and easily generated and comprehendible [24]. The game goals and learning goals need to be one and the same. For example, in *Angry Birds*, a famous mobile interface game the goals are clear and consistent. The pigs are to be destroyed by hitting the birds at correct angle and correct speed. This stays the same throughout the game but the introduction of different games throughout the game explains different subgoals and strategies.

- **Community and collaboration**
  Cooperative learning environments where learners' work in collaboration to achieve learning goals have shown improvement in the performance of the users. One of the advantages of using mobile devices for gaming is the benefit they offer for community and collaboration [24]. People enjoy playing with live players than with computer opponents, no matter how intelligent the games are. People tend to like and feel comfortable with the people they have played games with, no matter how badly they lost. Playing in collaboration fosters trusts among team members and promotes teamwork.

- **Context**
  The users' presence, location, expectations should all be taken into consideration during development. Context awareness plays an important role in the success of the games. Users respond differently in different situations and environments. If the context of the user is considered during the development phase, then the user can connect to the game in a better way [24]. Figure 3[24] illustrates considerations for context-are mobile games.

Presence	Who is playing the game?
Location	Where are learners playing the game?
Activity level	How active are they?
Actions	What are they doing?
Intentions	What will they do next? Where will they be?
Changes	What changes are they making, and where?
Objects	What objects are they using? What objects could they use?
Extents	What can they see? How far can they reach?
Abilities	What can they do?
Sphere of influence	Where can players make changes and how?
Expectations	What do players want the game to be able to do?

(Adapted from Cutwin and Greenberg 1996)

Proposed Educational Game

The proposed game will demonstrate the concepts of Java programming language to the players. The actions and the flow of the game will reinforce the concepts of the programming language. The game will cover the concepts of classes, objects, encapsulation, loops, inheritance and polymorphism. Clarity of these concepts can help players build their further knowledge in the programming language.The proposed game maps the animal kingdom to the object-oriented design principles. The user is given a choice of animals; he has to choose one animal which becomes his mascot and the game will be presented to the user from the aspect of the chosen animal. Animal kingdom can be classified into different classes of animals, where all the animals in one class have one or more same characteristics. This concept is used to teach the concept of classes in Java. The user is then presented with a sample code and is shown the instance variables and methods representing the characteristics and behavior of the animal he has chosen, to explain the concept with clarity. The user can then take a small quiz to test his knowledge of the concept. The concept of inheritance requires the user to select more than one animal, to show how the child class has properties of both parent classes and some of its own properties.The target users can be characterized as "casual gamers" and users who are interested in learning programming. Many of the users, however, may know some programming in Java, or may know a little of another programming language, or maybe have taken a class or two but need greater understanding. This game is intended to help you learn Java programming in a fun, clear way. the game can be played by the kids and gamers of any age to

learn the concepts of object-oriented programming, question arises that can kids learn programming? In his book "*Java Programming for Kids, Parents and Grandparents" author* Yakov Fain states that it is definitely possible to teach kids programming by relating programming concepts to the real-world objects. According to Yakov [25]:

- Most of the programming tasks require minimal knowledge of arithmetic and algebra skills. To start programming, a kid needs to understand what x = y+2 means and the concept of an IF statement.
- Kids develop the abstract reasoning abilities by the fourth-fifth grade, and they also easily perform such tasks as browsing the Web, downloading and installing software.
- Kids learn much faster than adults, but they do not have "previous programming experience", which may actually be a good thing, because they do not have to switch from a procedural to object-oriented way of thinking.
- Programming sessions and ways with kids should be embedded with graphics and should have small activities, by which they feel rewarded.

Hence, the inclusion of kids in target users can be justified. Similarly, users of all age can learn to program in Java, with this programming-oriented game.

Why Andriod?

- **Huge Market:** Recent statistics show that, Android captures 52% of the Smartphone market across the world. There are more Android devices in use world-wide than iPhone or Windows phone [26].
- **Open Source:** Android is open source i.e. Android Software Developer Kit (SDK) can be downloaded for free, and the development can be done on both Windows and Macintosh. Whereas for Android's strongest competitor, iOS development for iPhones' and iPads requires a Macintosh computer for development and the development SDK must be purchased at a high cost from Apple Inc.
- **Submission of app in the market:** It is far more expensive to submit apps in Apple store as compared to Android market. Also, any app that functions correctly and does not violate any terms can be placed in the Android Market.
- **Multiple market access:** Android submissions can be made to multiple app stores easily, which means easier access to markets like India, China and Korea. This can play a vital role in the success of an app and developer [26]..
- **Testing** – With Android it is very easy to test an app on different devices. An app can be packaged and sent over for testing purposes.

All these factors played an important role in deciding the platform for development. I have used PhoneGap, which uses native HTML5 and JavaScript for development, which is a good compliment to the Android SDK.

Scenario

Primary actor: A kid interested in computers and learning programming, looking for good educational games.

Description: John is an eighth-grade student. He likes computers and electronics since a very young age. He is interested in learning programming languages and wants to be a computer engineer in future. On his fifth birthday, his father bought a computer for him. He was very happy and played different games on the computer. He is a good learner and learns every game quickly. Over the years his interest in games has increased and he has been spending a lot of time playing different types of games. He is also a sharp student and plays games in his leisure time. He recently got an Android tablet as a Christmas gift. He enjoys playing different types of games on the tablet and keeps himself updated with the latest games in the market. He has played some educational games but doesn't enjoy them as his other games. He thinks that they are quizzes oriented and lose the game characteristics.

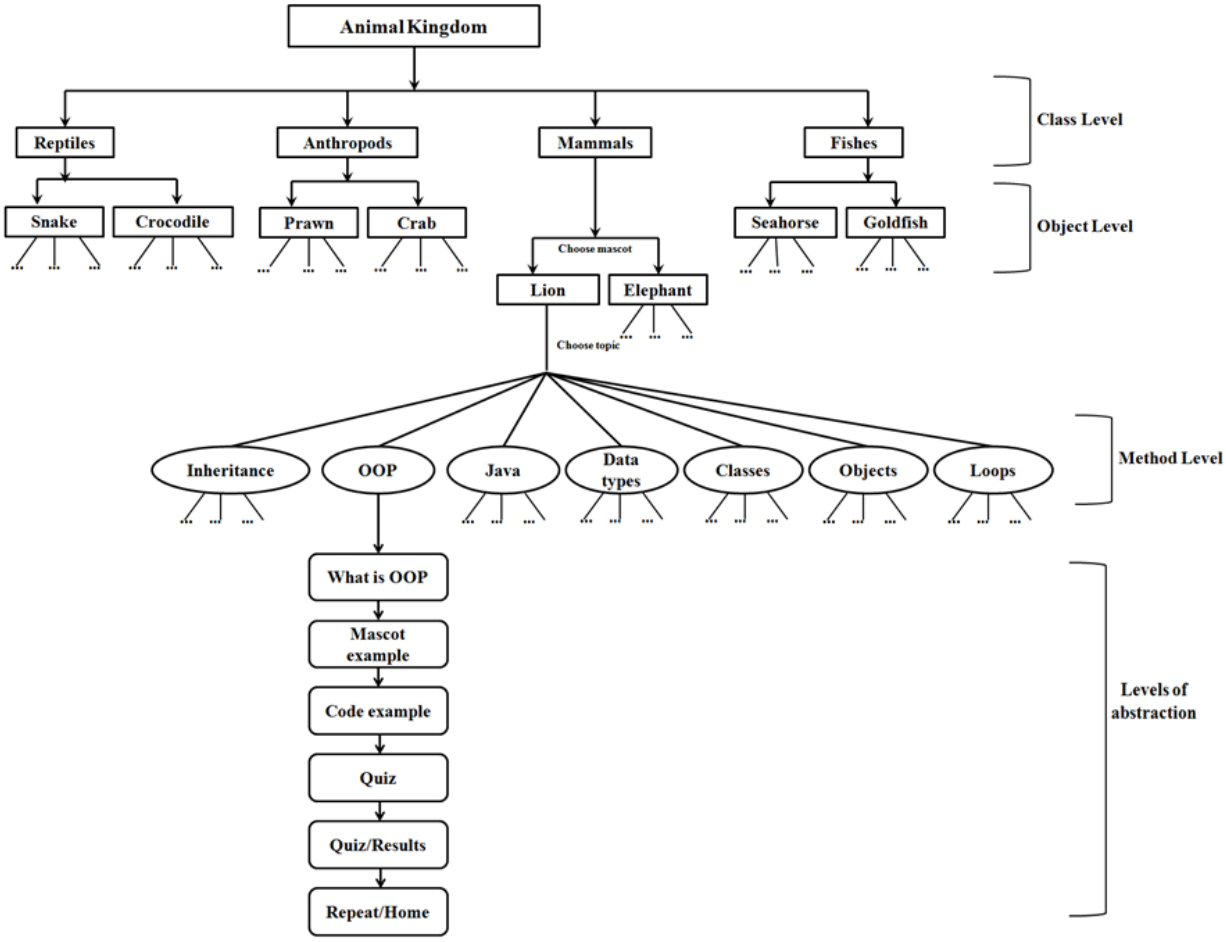
Event: When he came across the new JavaZOO game, he was not very excited, as his experience with educational games has not been great. He has some programming experience with other programming languages but he is new to Java. He plays the game on his Android tablet and sometimes on his father's phone. After downloading Java ZOO, he reads some information on how to play the game and what the game is about, and then starts to play the game. He has to choose an animal or a mascot from the list so that he can map the concepts of object-oriented programming to the real-world objects. He chooses lion as his mascot, and then chooses to learn about objects. The game gives him a brief description of what is an object in terms of object-oriented programming and then how is lion an object; with its state and behavior. He is then shown a sample code in the context of state and behavior of lion, after that he is given a summary and then he can take a small quiz to test his knowledge. He is thrilled about the game made him learn an important concept of programming language. He further plays the game and covers all the other sections provided in the game. It was a nice experience for him as he moves in the game taught him the concepts. He is very interested in the game and likes to play it again and again.

4. Implementation

In October 2011, Adobe acquired Nitobi enabling the team to focus solely on the PhoneGap project and continue its work on efficient expressive design and development across devices. Adobe will continue to host the PhoneGap online community as well as the PhoneGap Build service, which is now in open beta.PhoneGap is an open source implementation of open standards. Nitobi was the original creator and is one of the primary contributors to the PhoneGap framework, but there is a vast global community that also contributes to the project, including many from IBM, RIM and Microsoft. PhoneGap framework allows developers to natively develop applications for all smartphones with a single codebase (JavaScript, HTML and CSS) by enabling a Foreign Function Interface (FFI) to an embedded WebView or Webkit on the device. PhoneGap currently supports development for the operating systems Apple iOS, Google Android, HP webOS, Microsoft Windows Phone, Nokia Symbian and RIM BlackBerry.

4.2 Development

4.2.1 Object-Oriented Design of JavaZOO



4.2.2 Paper prototyping

For developing the prototype of JavaZOO, initially, the paper prototype was designed. The requirements were gathered and analyzed based on Evolutionary Prototyping (EP). EP allows a continuous refinement of the system and is based on the acknowledgement that designers may not understand all the requirements and will build on those well-understood requirements while adding features as they understand the requirements more fully. Initially all the features that are needed to be included in the system are designed and analyzed based on the existing systems and a model was proposed.

4.2.3 Process

While developing JavaZOO a complete development lifecycle was followed. The project underwent series of iterations of analysis, design, coding and testing during development.

- Requirement Analysis: In JavaZOO requirement analysis helped in investigating the target users and how can learning in form of a game aid their knowledge of programming. Our analysis identified key features of the game and the information that we want to include in the game. The platform of development was studied and further progress was decided.

- Specification Building: In JavaZOO, all the specifications of the game were documented, the development platform was documented and further progress in design phase was decided. The specifications mainly included the details about language that will be used in the game to teach the concepts of programming, how will the programming concepts be mapped to the game, how the balance between text and graphics will be maintained, which programming concepts will be included in the game, how will code snippets be included etc.
- Design & Development: Design played an important role in this project, as educational games can lose their purpose if proper balance is not maintained in the amount of information, text and graphics to be included. We developed solutions for all the concerns that were raised during prototyping. In this phase, we decided how the concepts of classes, objects, loops, encapsulation and inheritance will be depicted. For example, after a user chooses a mascot, if he/she chooses classes as the topic, he/she will be given a small introduction of what are classes in object-oriented programming, he/she will then be taken to a classification of animal kingdom where he/she can see which class does his mascot belongs to, then he/she will be shown a small code snippet of a class where his/her chosen mascot will be one of the objects of that class type (this will help in relating the programming concept to the real world), he/she will then be given a summary of what classes are why are they important and then he/she can take a small quiz for testing what he has learned so far. Finally, the quiz will be graded and answers to quiz questions will be presented. In the design phase, all the details of every topic that would be included in the game were designed for helping in planned development.
- Content Writing: In this phase, all the text or details that must be included in the project are documented. In JavaZOO, the content writing was very important as with the limited screen size of mobile devices, too much of text can make the user lose interest in the game and can also be unreadable. Hence, it was very important to carefully review the content and present definitions of all the concepts, so that they were informative and according to the screen size.
- Coding: In this phase, the focus was on coding a solution that was developed in the design phase. The code was written using HTML5 and JavaScript on Eclipse Integrated development environment (IDE) with PhoneGap as the platform basis. Appropriate graphics included to make the game aesthetically pleasing.
- Testing: This is the phase where the developed code should undergo testing to see whether the code is validated according to the requirements and design class model. However, when a system passes the initial testing, it doesn't mean it's right, it only means it passes the test. Therefore, incremental testing and development are essential to achieving a robust system. In JavaZOO, the system was tested for all the specifications that were documented in the specifications building phase and other tests to ensure application robustness.

5. Results and Analysis

The primary objectives of the study were to answer the following research questions:

- How educational games can prove as effective learning and teaching technique?
- Is the proposed game serving the purpose of teaching the programming concepts to the novice users?
- How effectively can the mapping of programming concepts and animal kingdom be designed in the form of a game?
- Investigate the issue of teaching programming through games.
- Investigate the affordances and constraints involved in developing an educational game that can explain object-oriented programming concepts in context to the real world, in a capacity that is easy to use, meets all the requirements and is engaging for the learners.
- Design an entertaining educational game with a challenge of maintaining education aspects and game aspects balanced.
- Determine if the developed prototype is acceptable from the user interface standpoint.
- Determine if the developed prototype is acceptable for teaching object-oriented programming concepts.

To address these research questions, a series of studies were conducted. JavaZOO was designed keeping in mind all these research questions. The developed game is tested with users using a questionnaire survey. As the target users of the game are not specific to any group of people, the study was conducted with the group consisting of novice users, young children, avid gamers, non-gamers, users with extensive object-oriented concepts knowledge. The results from these

surveys provided information about how the users perceived the developed game and if the proposed application could explain to them, the concepts of object-oriented programming. The users played the game and gave their opinions through an online survey. The user survey consisted of three sections. The first section had questions regarding the system usability. The second section had questions regarding information quality and the third section had open-ended questions. A likert-type scale was used for the questionnaire.

5.1 Results of Usability for JavaZOO

The results from this section of the survey showed the user's perception of the system from the usability point of view. Data were collected on a six-point Likert scale, with "strongly disagree" being the lowest and "strongly agree" being the highest value for the responses.Most of the users indicated that they were satisfied with the system usability. Over 90% of the users reporting that the agreed that the system was easy to use. Eighty-six% agree it was easy to learn and 93% agree that they could relate object-oriented programming concepts to the animal kingdom and understand them better. Seventy-one% of users reported that they feel comfortable using the system and 62% of the users reported that the layout of the interface was visually appealing.

Results of information quality for JavaZOO. This section of the questionnaire survey reported the user's perception of the system from the information quality point of view. Data were collected on a six-point Likert scale, with "strongly disagree" was being the lowest and "strongly agree" being the highest value for the responses.

There were 3 open-ended questions in the questionnaire. Participants made specific comments regarding the system in this section of the questionnaire. The users reported suggestions to improve the game as follows: improving the layout and colors, adding audio, interaction and more topics. The users also reported aspects of JavaZOO that were positive. Some of these include the following: nice graphics; easy to use and gives a conceptual understanding of object-oriented concepts; It was entertaining along with educational; It was easy to use; and It had great graphics and was a great way to explain computing concepts.

Conclusion

In this research project, we proposed JavaZOO to demonstrate the concepts of Java programming language to the players. The actions, flow of the game and activities were created to reinforce theoretical programming concepts such as classes, objects, encapsulation, loops, inheritance and polymorphism. Our aim with this game was to increase retention of CS students in the majors and providing more clarity of these concepts can help support strong computing students as they play JavaZOO to increase their knowledge of the 0Java programming language. The proposed game maps the animal kingdom to the object-oriented design principles. From the evaluation of this project we have found that this application was highly satisfactory to the users of the system and effectively introduced them to computing concepts. Our plans for this project is to continue to test it with a larger population of students and refine this and other games to in effect game more of the introductory Computer Science Curriculum and keep the highly-talented students that we have recruited to this discipline.

References

[1]. Spohrer, J. and Soloway, E. (1986) Novice mistakes: are the folk wisdoms correct? Commun. ACM, 29, 7, 624–632

[2]. Jens J. Kaasbøll (1999) Exploring didactic models for programming. Tapir pp. 195-203.

[3]. Rogalski and Samurçay (1990) Acquisition of programming knowledge and skills. In Hoc, J.-M (ed.) Psychology of programming. London, Academic Press

[4]. du Boulay, Benedict and O'Shea, Tim and Monk, John (1999) The black box inside the glass box: presenting computing concepts to novices.International Journal of Human-Computer Studies, 51 (2). pp. 265-277. ISSN 1071-5819.

[5]. Siti Hafizah, Ab Hamid and Leong Yu Fung (2007) Learn Programming by Using Mobile Edutainment Game Approach. The First IEEE International Workshop on Digital Game and Intelligent Toy Enhanced Learning. pp 1-4

[6]. Shiffman, D. (2010, October 28). Interview on programming, computation, data, and why everyone should learn programming [Video file]. Retrieved from http://flowingdata.com/2010/10/28/why-everyone-should-learn-programming/

[7]. G. M. Seddon (1978) The Properties of Bloom's Taxonomy of Educational Objectives for the Cognitive Domain. Review of Educational Research Vol. 48, No. 2 (Spring, 1978), pp. 303-323

[8]. Fang, Z. Illustrations, text, and the child reader. What are pictures in children's storybooks for? Reading Horizons, 37, (1996), 130-142.

[9]. Ricci, K., Salas, E., Cannon-Bowers, J. A. Do computer-based games facilitate knowledge acquisition and retention? Military Psychology, 8(4), (1996), 295-307.

[10]. Pierfy, D. A. Comparative simulation game research. Simulation & Games, 8, (1977), 255-268.

[11]. Druckman, D. The educational effectiveness of interactive games, In D. Crookall & K. Arai (Eds.), Simulation and gaming across disciplines and cultures: ISAGA at a watershed, (1995), 178-187.

[12]. Booth, S. (1992) Learning to program: a phenomenographic perspective. ACTA Univ. Gothenburg studies in educational science 89.

[13]. Caillois, R. Man, play, and games. New York, Free Press (1961).

[14]. Druckman, D. The educational effectiveness of interactive games, In D. Crookall & K. Arai (Eds.), Simulation and gaming across disciplines and cultures: ISAGA at a watershed, (1995), 178-187.
[15]. Fang, Z. Illustrations, text, and the child reader. What are pictures in children's storybooks for? Reading Horizons, 37, (1996), 130-142.
[16]. Freeman, T. Monitoring Devices in Games. Game Developer.(1997)**,** 22-31.
[17]. Garris, R, Ahlers, R. Games, motivation, and learning: A research and practice model. Simulation & Gaming, 33, (2002), 441-467.
[18]. Gordon, A. Games for Growth. Sc.Research Associates, Inc., Palo Alto, California (1970).
[19]. Malone, T. W. What makes things fun to learn? A study of intrinsically motivating computer games. Cognitive and Instructional Sciences Series, CIS-7, SSL-80-11, Palo Alto Research Center, Palo Alto (1980).
[20]. McGonigal, J. "Reality is Broken:Why Games Make Us Better and How They Can Change the World". Penguin Press (2011).
[21]. Pierfy, D. A. Comparative simulation game research. Simulation & Games, *8*, (1977), 255-268.
[22]. Quinn, C.N. Pragmatic Evaluation: Lessons from Usability. 13th Annual Conference of the Australasian Society for Computers in Learning in Tertiary Education, Australasian Society for Computers in Learning in Tertiary Education, Adelaide, (1996).
[23]. Ricci, K., Salas, E., Cannon-Bowers, J. A. Do computer-based games facilitate knowledge acquisition and retention? Military Psychology, *8*(4), (1996), 295-307.
[24]. Thomas, S., Schott, G. and Kambouri, M. Designing for Learning or Designing for Fun? Setting Usability Guidelines for Mobile Educational Games. Proceedings of MLEARN Learning with Mobile Devices, London, (2003).
[25]. Yakov Fain book *"Java Programming for Kids, Parents and Grandparents"*
[26]. Ricknas, M (2011). Samsung Becomes Biggest Smartphone Vendor, as Android's Market Share Grows, PCWorld. http://www.pcworld.com/article/243861/samsung_becomes_biggest_smartphone_vendor_as_androids_market_share_grows.html.
[27]. Android Architecture retrieved fromhttp://developer.android.com/guide/basics/what-is-android.html.
[28]. Hardesty, Larry (August 19, 2010). "The MIT roots of Google's new software". MIT News Office. http://web.mit.edu/newsoffice/2010/android-abelson-0819.html.
[29]. On the Shoulders of Giants, Google.. Retrieved August 10, 2010. http://appinventor.googlelabs.com/about/
[30]. The web site provides many links to explain web design and application languages like HTML and CSS (http://www.w3.org/standards/webdesign/htmlcss.html).
[31]. The web site provides many links to explain various types of fidelity (http://www.usabilityfirst.com/glossary).